SNEED: Enhancing Network Security Services Using Network Coding and Joint Capacity

Salah A. Aly Nirwan Ansari H. Vincent Poor

***Abstract*—Traditional network security protocols depend mainly on developing cryptographic schemes and on using biometric methods. These have led to several network security protocols that are unbreakable based on difficulty of solving untractable mathematical problems such as factoring large integers.**

In this paper, *Security of Networks Employing Encoding and Decoding* (SNEED) is developed to mitigate single and multiple link attacks. Network coding and shared capacity among the working paths are used to provide data protection and data integrity against network attackers and eavesdroppers. SNEED can be incorporated into various applications in on-demand TV, satellite communications and multimedia security. Finally, It is shown that SNEED can be implemented easily where there are k edge disjoint paths between two core nodes (routers or switches) in an enterprize network.

I. Introduction

Internet Service Providers (ISP) and Internet Traffic Engineering (ITE) aim to provide fast, reliable, quality of demands, and differentiated services for demanding users. Such services can be deployed at the IP, physical, and application layers. Several security schemes and network protection strategies have been proposed during the last two decades to protect operational networks against link failures, node attacks, increased overhead and congestion. The goal in this paper is to provide novel strategies for network security services against attacks and eavesdroppers by deploying network coding and shared capacity. The strategies can be deployed to protect network traffics between core nodes such as routers or switches.

Protection of communication networks against network attacks and failures are essential to increase robustness, reliability, and availability of the transmitted data. The attacks may also occur at various network layers, including the physical, IP, or application layers [13], [16]–[18]. Also, network attacks can occur due to vulnerable network configurations or due to transmitting insecure data. These problems have received significant attention from researchers and practitioners, and a large number of techniques have been introduced to address such problems. In the other side, traditional network security schemes depend mainly on developing cryptographic protocols or on using biometric methods. Essentially, cryptographic protocols are considered unbreakable based on difficulty of solving mathematical problems such as factoring large integers [17], [18].

Network coding is a powerful tool that has been recently used to increase the throughput, capacity, and performance of communication networks. Information theory aspects of network coding have been investigated in [1], [7], [8] and [20]. It certainly can offer benefits in terms of energy efficiency, additional security, and delay minimization. Network coding is used to detect adversaries [10] and to protect packets against network attackers and injectors [5], [9], [12]. Network coding can be also used to enhance security and protection [12], [14].

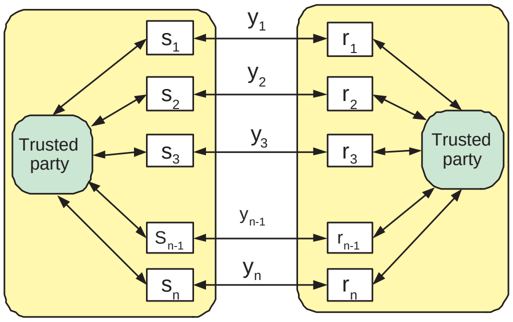

Fig. 1. n links are shared between sets of senders and receivers. One link is used for data integrity and message authentication. The trusted party can be a router (or switch) to send and receive messages.

In this paper, we propose an approach for light-weight network security that is based on network coding. We develop a scheme called **SNEED**, Security of Networks Employing Encoding and Decoding, in order to protect transmitted data between sets of senders and receivers. For one path that has been attacked (eavesdropped) between a sender and receiver, one backup path is provided, in which it will carry encoded data from sources to receivers.

II. Network Model and Assumptions

In this section, we describe the network model and basic assumptions. The node can be a router, switch, or an end terminal depending on the network model and transmission layer.

A. Network Model

i) Let $\mathcal{N}$ be a network represented by an abstract graph $G = (\mathbf{V}, E)$, where $\mathbf{V}$ is a set of nodes and E is a set of undirected edges. Let $S = \{s_1, s_2, \ldots\}$ and $R = \{r_1, r_2, \ldots\}$ be sets of independent sources and destinations, respectively. The set $\mathbf{V} = V \cup S \cup R$ contains the relay nodes, sources, and destinations as shown in Fig. 1. Assume for simplicity that $|S| = |R| = n$, and hence the number of sources is equal to the number of receivers. Assume also that the senders (receivers) are connected by a super sender (receiver).

ii) A connection path L_i is a set of edges connected together with a starting node (sender) and an ending node (receiver). The paths $L = \{L_1, L_2, \ldots\}$ carry data from the sources to the receivers. Connection paths are link disjoint and provisioned in the network between senders and receivers. All connections have the same bandwidth, or otherwise a connection with a high bandwidth can be divided into multiple connections, each of which has the unit capacity.

iii) Every sender s_i will send a message m_i^ℓ to the receiver r_i at time t_δ^ℓ in the round ℓ in the cycle δ, for all $1 \leq i \leq n$. A receiver r_i receiving a message m_i^ℓ is able to detect whether the message has been altered by using any authentication or signaling protocols. Further details regarding this model, definition of the working and backup paths, and the normalized capacity can be found in [2] and [3].

iv) Enc_k and Dec_k represent the encryption and decryption algorithms with a shared symmetric key k, respectively.

B. Senders and Receivers Packets

Every sender s_i prepares a packet $packet_{s_i \rightarrow r_i}$ sent to the receiver r_i. The packet contains the sender's ID_{s_i}, data m_i^ℓ, and time for every round and cycle t_δ^ℓ. There are four types of packets that carry the data:

i) **Plain Packets.** Packets sent without network coding or encryption, in which the sender does not require to perform any coding or encrypting operations. For example, in case of packets sent without coding, the sender s_i sends the following packet to the receiver r_i:

$$packet_{s_i \rightarrow r_i} := (ID_{s_i}, m_i^\ell, t_\delta^\ell) \tag{1}$$

ii) **Encoded Packets.** Packets sent with encoded data without encryption, in which the sender requires to perform other sender's data. For example, in case of packets sent with encoded data, the sender s_i sends the following packet to receiver r_i:

$$packet_{s_i \rightarrow r_i} := (ID_{s_i}, \sum_{s_j \in \mathcal{S}} m_j^\ell, t_\delta^\ell), \tag{2}$$

where $\mathcal{S}$ is the set of sources sending plain messages.

iii) **Encrypted Packets.** Assume there is a shared symmetric key between a sender s_i and receiver r_i. In this case the sender s_i will send the packet $packet_{s_i \rightarrow r_i}$ as follows:

$$packet_{s_i \rightarrow r_i} := (ID_{s_i}, Enc_{k_i}(m_i^\ell), t_\delta^\ell). \tag{3}$$

This packet carries an encrypted message without encoding.

iv) **Encoded and Encrypted Packets.** Packets sent with encoded data with encryption, in which the sender needs to protect other senders' data. For example, in case of packets sent with encoded encrypted data, the sender s_i sends the following packet to receiver r_i:

$$packet_{s_i \rightarrow r_i} := (ID_{s_i}, \sum_{s_j \in \mathcal{S}} x_j^\ell, t_\delta^\ell). \tag{4}$$

The value $y_i = \sum_{j=1, j\neq i}^{n} x_j^\ell$, where $x_j^\ell = Enc_{k_j}(m_j^\ell)$, is computed by every sender s_i, in which it is able to collect the data from all other senders and encode them by using the XORed operations.

C. Attackers Model

We represent an attacker model as follows. There are two types of attacks in which the network security services must overcome: Active (intruders) and passive (eavesdroppers) attackers [19]. We also assume that t different attackers have access to t channels among all n channels $L_1, L_2, \ldots, L_n$ at a certain round time, for $t \geq 1$. This is similar to an attacker accessing t channels. Multiple attackers which attack the same channel are represented by one attacker.

i) The passive attacker is able to eavesdrop on the transmission between the senders and receivers. A passive attacker such as an eavesdropper should not learn any information even if it can have a copy of it.

ii) The active attacker can modify or fabricate messages throughout a cycle. This occurs by injecting new data (coefficients) at the relay nodes of data sent by the sources. This can occur also over the shared links. In addition, an active attacker will not be able to change or fabricate information, and affect the system resources due to the network security strategies.

This attacker model is similar to the attacker model described for wiretapping channels as stated by many authors [6], [20].

III. Data Security Against A Single Attacked Path By Using Shared Keys And Network Coding

In this section, we consider the case of a single *active* attacker, i.e., $t = 1$. We will assume that there are shared symmetric keys between the senders and receivers. Also, the receivers are able to detect messages that have been modified by the attackers using hashing functions like MD5 or SHA-1; fabricated messages can be detected by using sequence numbers between the senders and receivers [17], [18].

Let k_i be a shared symmetric key between s_i and r_i. This key can be distributed by using a Trusted Third Parity (TTP). In this case, the senders exist in a secure domain as well as the receivers. Let x_i be the encrypted message from the sender s_i to the receiver r_i by using the shared key k_i. Thus,

$$x_i^\ell = Enc_{k_i}(m_i^\ell) \tag{5}$$

A. Encoding Operations: The encoding operations are done as follows. At every round time, $n-1$ senders will send their own data with full capacity over $n-1$ paths that are established from the sources to the destinations. Also, these $n-1$ sources will exchange their data with exactly one source node s_i that will send the Xored encoded data over a shared link L_i. This process is explained in Eq. (6) in Table 1.; and we call it **(SNEED)** against a single attacked path (SAP). The data is sent in rounds for every cycle. Also, we assume that the attacker can affect only one path throughout a cycle,

TABLE I
THE ENCODING OPERATIONS OF **SNEED**

	round time cycle 1							...	...
	1	2	3	...	j	...	n	...	...
$s_1 \to r_1$	y_1	x_1^1	x_1^2	...	x_1^j	...	x_1^{n-1}	...	...
$s_2 \to r_2$	x_2^1	y_2	x_2^2	...	x_2^j	...	x_2^{n-1}	...	...
$s_3 \to r_3$	x_3^1	x_3^2	y_3	...	x_3^j	...	x_3^{n-1}	...	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	...
$s_j \to r_j$	x_j^1	x_j^2	x_j^3	...	y_j	...	x_j^{n-1}	...	...
$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$	...	...
$s_n \to r_n$	x_n^1	x_n^2	x_n^3	...	x_n^{j-1}	...	y_n	...	...

(6)

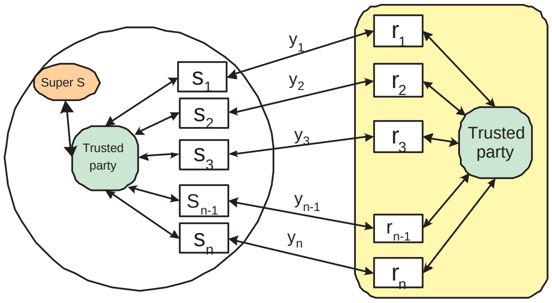

Fig. 2. n shared links between a set of senders and a set of receivers. One link is used for data integrity and authentication. A path is a set of links connecting a set of nodes (routers or switches). A software management to solve a commodity problem can be used to provision a set of link disjoint paths between two core nodes in a given network topology.

but different paths might suffer from different *active* attackers throughout different cycles. In this case, throughout of one cycle consists of n rounds, y_j's for $1 \leq j \leq n$ are defined over $\mathbf{F}_2$ as

$$y_j = \sum_{i=1}^{j-1} x_i^{j-1} \oplus \sum_{i=j+1}^{n} x_i^j. \tag{7}$$

The senders send packets to the set of receivers in rounds. Every packet initiated from the sender s_i contains ID_{s_i}, data $x_{s_i}^\ell$, and a round t_δ^ℓ. For example, the sender s_i will send the encrypted $packet_{s_i \longrightarrow r_i}$ as follows.

$$packet_{s_i \longrightarrow r_i} = (ID_{s_i}, x_{s_i}^\ell, t_\delta^\ell). \tag{8}$$

Also, the sender s_j will send the encoded encrypted data y_{s_j} as

$$packet_{s_j \longrightarrow \longrightarrow r_j} = (ID_{s_j}, y_{s_j}, t_\delta^\ell). \tag{9}$$

We ensure that the encoded data y_{s_j} is varied per one round transmission for every cycle. This means that the path L_j is dedicated to send only one encoded data y_j and all data $x_j^1, x_j^2, \ldots, x_j^{n-1}$.

The data transmitted from the sources do not experience any round time delay. This means that the receivers will be able to decrypt the received packets online and immediately recover the attacked data. In Eq. (6), the *active* attacker can break only one message per one attacked working path. A generalization of this scheme is presented in Section V where the *active* attacker(s) has access to t multiple channels simultaneously.

***Lemma** 1:* The normalized network capacity according to Eq. (6) is $(n-1)/n$.

Proof: The proof comes from the fact that only one encoded packet is sent over one channel throughout every round per cycle. Therefore, there are $(n-1)$ plain packets sent over n channels. ■

B. Decoding and Data Integrity Operations: The decoding operations are done as follows. Every receiver r_i will receive a message x_i^ℓ over the link L_i. Once the attack occurs at a link L_i, then the receiver that receives y_ℓ over the path L_ℓ will be used for data integrity and recovery.

In case the attacker modifies the message, the receiver r_i will know about the modified message by using MD5 hashing function, so r_i will ask other receivers to send their messages to recover the modified message.

$$x_i^\ell = y_j \oplus \sum_{h=1, h \neq i}^{n} x_h^{j-1} \oplus \sum_{h=j+1, h \neq i}^{n} x_h^j. \tag{10}$$

The receiver r_i will decrypt the message x_i^ℓ by using its symmetric key k_i, i.e., $m_i^\ell = dec_{k_i}(x_i^\ell)$.

IV. **SNEED** WITHOUT SHARING SYMMETRIC KEYS

In this section, we propose to use network coding to secure the traffic between the senders and receivers against *active* attackers, where no symmetric keys are shared. Assume a network with n connections shared between n senders and receivers. We will assume that every sender will be able to combine packets from other receivers to hide its own data. For example, the sender s_i will send the encoded message y_i to the receiver r_i over the link L_i.

We will design a security scheme by using network coding against an entity which can not only copy or listen to the message, but also can fabricate new messages or modify the current ones. In this model we do not assume pre-shared secret keys between the senders and receivers. Also, the message is still secured against attack's fabrication and modification.

A. Network Security Codes and **SNEED**

We will define network security codes for **SNEED** and study their properties. We assume there is a super sender S that sends n different messages over disjoint paths to n receivers as shown in Fig. 2. Furthermore, the receivers can communicate with each other by using a trusted party. The goal is to provide collaborative security for the n messages against eavesdroppers, intruders, and malicious attackers. We will develop **SNEED** over the binary field, hence the fastest encoding and decoding operations are used.

One useful application of **SNEED** is the case of sending multimedia and TV streams over a public network as in the

TABLE II
BEST KNOWN **SNEED** CODES OVER $\mathbf{F}_2$ [11]

n	m	code	type
7	3	$[7,4,3]_2$	Hamming code
10	4	$[10,6,3]_2$	Linear code
15	4	$[15,11,3]_2$	Hamming code
19	7	$[19,12,3]_2$	Extension construction
23	8	$[23,15,3]_2$	Extension construction
25	5	$[25,20,3]_2$	Linear code
31	5	$[31,26,3]_2$	Hamming code
39	8	$[39,31,3]_2$	Extension construction
47	9	$[47,38,3]_2$	Extension construction
63	6	$[63,57,3]_2$	Hamming code
71	8	$[71,63,3]_2$	Matrix construction
79	9	$[79,70,3]_2$	Extension construction
95	10	$[95,85,3]_2$	Extension construction
127	7	$[127,120,3]_2$	Hamming code

Internet. Such streams must be processed online with fast encoding and decoding operations, in addition to the security operations. Let d be the minimum distance defined as in the notion of error correcting codes [4], [11], [15].

***Definition** 2 (**SNEED**):* An $[n,k,d]_2$ network security code is a k-dimensional subspace of the space $\mathbf{F}_2^n$ that secures k information symbols (messages) by mapping them into n mixed symbols, and can recover from upto $d-1$ compromised (attacked) channels. Furthermore, the code is generated by a nonsystematic matrix G of size $k \times n$ defined over $\mathbf{F}_2$.

$$G = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{21} & \cdots & g_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{pmatrix}_{k \times n} \tag{11}$$

We will use the nonsystematic classical binary error correcting codes in the construction of **SNEED** [11], [15]. The encoding scheme of such codes is given by

$$\begin{array}{c||cccc} & L_1 & L_2 & \cdots & L_n \\ \hline\hline s_1 & g_{11}m_1 & g_{12}m_1 & \cdots & g_{1n}m_1 \\ s_2 & g_{21}m_2 & g_{22}m_2 & \cdots & g_{2n}m_2 \\ & \vdots & \vdots & \cdots & \vdots \\ s_k & g_{k1}m_k & g_{k2}m_k & \cdots & g_{kn}m_k \\ \hline\hline & y_1 & y_2 & \cdots & y_n \end{array} \tag{12}$$

The encoding message y_j, for $1 \leq j \leq n$, is defined by

$$y_j = \sum_{i=1}^{k} g_{ij} m_i \tag{13}$$

***Lemma** 3:* The normalized capacity of the network utilizing **SNEED** is given by k/n.

Proof: By the definition of the network security code, there are $n-k$ redundant symbols that are used to recover from up to $d-1$ attached channels. Therefore there are k working paths that will carry k source data. The result is a consequence by dividing by the total number of channels n. ■

Table II presents the best known **SNEED** for certain number of channels defined over $\mathbf{F}_2$.

We have $n-k$ lockers' channels, in which they carry redundant data, in this model. From the proposed code construction, we ensure that $t = d-1 \leq n-k$, where t is the number of compromised (attacked) channels. This is actually a direct consequence of the Singleton bound [11], [15].

B. Decoding Operations of SNEED

The decoding operations at the receivers side are guaranteed once a system of t linearly independent equations is established in t unknown variables. Let t attackers can access t disjoint channels and alter the transmitted messages. We assumed that the receivers are located in a trusted domain, therefore they can trust and exchange protected messages with each others.

The system can be solved by using, for example, Gauss elimination method. By definition of **SNEED**, the matrix $\mathbf{G}$ has dimension of k. Furthermore, the receivers will know the number and position of the channels that have been attacked. In this case the decoding operations are achieved by using the well known decoding methods for erasure channels, see [11].

The following example illustrates the proposed model.

Example 4: Assume we have a connection L_i between a sender s_i and a receiver r_i for $i = 1,2,3$ and 4. Furthermore, the channel L_4 is used as a lock (redundant) path. Without loss of generality, we can assume that the four senders send

$$\begin{aligned} y_1 &= m_1 \oplus m_2 \\ y_2 &= m_2 \oplus m_3 \\ y_3 &= m_1 \oplus m_3 \\ y_4 &= m_1 \oplus m_2 \oplus m_3 \end{aligned} \tag{14}$$

We also assume that the attacker affects only one of channels L_1, L_2 or L_3. In this example the **SNEED** can be stated as follows. There are three working paths and one lock path. The security scheme is given by

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \qquad \begin{array}{c||cccc} & L_1 & L_2 & L_3 & L_4 \\ \hline\hline s_1 & m_1 & 0 & m_1 & m_1 \\ s_2 & m_2 & m_2 & 0 & m_2 \\ s_3 & 0 & m_3 & m_3 & m_3 \\ \hline\hline T & y_1 & y_2 & y_3 & y_4 \end{array} \tag{15}$$

If the channel L_2 is compromised, then the decoding can be done by using Gauss elimination method over the channels L_1, L_3 and L_4. Adding y_1, y_3 and y_4 will give m_1, then substituting in y_1 and y_3 will give m_2 and m_3.

V. SNEED OVER HIGHER FINITE FIELDS

In this section, we study security of networks employing encoding and decoding, **SNEED**, against multiple link attacks. We propose **SNEED** over a finite field with q elements to achieved this goal. In this scheme is an extension of the scheme presented in Section III, where the encoding and decoding operations are defined over the binary field. Assume t be the number of compromised channels. One can design a matrix $\mathcal{G}$ over $\mathbf{F}_q$ such that $k = n - t$ paths will carry secure data. Let a be a primitive element in $\mathbf{F}_q$. The matrix $\mathcal{G}$ is defined by

$$\mathcal{G} = \begin{pmatrix} 1 & a & a^2 & \dots & a^{n-1} \\ 1 & a^2 & a^4 & \dots & a^{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a^k & a^{2k} & \dots & a^{kn} \end{pmatrix} \tag{16}$$

The matrix G has rank $k = n - t$. Clearly a finite field with $q > n-t+1$ is sufficient for the encoding and decoding operations. The encoding operations are done by using the following encoding scheme.

$$\begin{array}{c||ccccc} & L_1 & L_2 & L_3 & \cdots & L_n \\ \hline\hline s_1 & x_1 & ax_1 & a^2x_1 & \dots & a^{1n}x_1 \\ s_2 & x_2 & a^2x_2 & a^4x_2 & \dots & a^{2n}x_2 \\ & \vdots & \vdots & \vdots & \dots & \vdots \\ s_k & x_k & a^kx_k & a^{2k}x_k & \dots & a^{kn}x_k \\ \hline\hline T & y_1 & y_2 & y_3 & \dots & y_n \end{array} \tag{17}$$

By this construction, if there are up to t attacked paths, then the system of $n - t \times n - t$ equations is solvable. This is due to the fact that the remaining matrix can be reduced to the Vandermond matrix [11, Chapter 4].

VI. Conclusion

Network coding as a promising tool offers benefits for enhancement and supplement of network security services. In this paper, we presented schemes for enhancing network security using network coding and joint capacities. We demonstrated the encoding and decoding operations of the proposed **SNEED** and showed that it can be deployed over a network with n senders and n receivers. Furthermore, **SNEED** is robust against active and passive network attacks. Our future work will include practical aspects of the proposed schemes.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Trans. Inform. Theory*, 46(4):1204–1216, 2000.

[2] S. A. Aly and A. E. Kamal. Network protection codes against link failures using network coding. In *Proc. IEEE GlobelComm '08, New Orleans, LA*, December 1-4 2008. arXiv:0809.1258v1 [cs.IT].

[3] S. A. Aly, A. E. Kamal, and A. Walid. Network protection design using network coding. In *Proc. 2009 IEEE International Workshop on Information Theory (ITW2010), Cairo, Egypt*, January 8-10, 2010. arXiv:1008.4264v1 [cs.IT].

[4] E. Ayanoglu, R. D. Gitlin Chih-Lin, and J. E. Mazo. Diversity coding for transparent self-healing and fault-tolerant communication networks. *IEEE Trans. on Communications*, 41(11), pages 1677-1686, November 1993.

[5] N. Cai and W. Yeung. Network error correction, part 2: Lower bounds. *Communications in Information and Systems*, 6:37–54, 2006.

[6] E. El-Rouayheb, S. Y. Soljanin. On wiretap networks II. In *Proc. IEEE ISIT 2007*, pages 551–555, Nice, France, July, 2007.

[7] C. Fragouli, J. Le Boudec, and J. Widmer. Network coding: An instant primer. *ACM SIGCOMM Computer Communication Review*, 36(1):63–68, 2006.

[8] C. Fragouli and E. Soljanin. Network Coding Applications, Foundations and Trends in Networking. *Hanover, MA, Publishers Inc.*, 2(2):135–269, 2007.

[9] C. Gkantsidis and P. Rodriguez. Cooperative security for network coding file distribution. In *Proc. IEEE INFOCOM*, Barcelona, Spain, April, 2006.

[10] T. C. Ho, B. Leong, R. koetter, M. Medard, M. Effros, and D. R. Karger. Byzantine modification detection in multicast networks using randomized network coding. In *Proc. IEEE ISIT*, 2004.

[11] W. C. Huffman and V. Pless. *Fundamentals of Error-Correcting Codes*. Cambridge University Press, Cambridge, 2003.

[12] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Medard. Resilient network coding in the presence of byzantine adversaries. In *Proc. IEEE INFOCOM*, Anchorage , AK , USA, April, 2007.

[13] J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. Chapman &Hall/ CRC, 2008.

[14] L. Lima, M. Medard, and J. Barrows. Random linear network coding: A free cipher. In *ISIT 06*, 2006.

[15] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1977.

[16] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL, 2001.

[17] B. Schneier. *Applied Cryptography*. 2nd edition, Wiley, New York, 1996.

[18] W. Stallings. *Cryptography and Network Security*. Pearson Prentice Hall, Inc., Fourth Edition, 2006.

[19] D. R. Stinson. *Cryptography, theory and practice*. CRC Press, 1995.

[20] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang. *Network Coding Theory*. Now Publishers Inc., Dordrecth, The Netherlands, 2006.